

```

;*****
; Copyright © [12/09/1998] Scenix Semiconductor, Inc. All rights reserved.
;
; Scenix Semiconductor, Inc. assumes no responsibility or liability for
; the use of this [product, application, software, any of these products].
; Scenix Semiconductor conveys no license, implicitly or otherwise, under
; any intellectual property rights.
; Information contained in this publication regarding (e.g.: application,
; implementation) and the like is intended through suggestion only and may
; be superseded by updates. Scenix Semiconductor makes no representation
; or warranties with respect to the accuracy or use of these information,
; or infringement of patents arising from such use or otherwise.
;*****
; Simple FSK modulator utilizing an artificial SINE wave generator
; and one PWM output
;
; Designed for use with Scenix DTMF demo boards
;
; Authors: Chris Fogelklou, Scenix Semiconductor Inc.
; Written: 98/12/09 to 98/12/09
;
; This program demonstrates a simple FSK modulation scheme. It contains
; no UART, so the baud rate of the FSK modulation is set by the external
; Terminal's UART. It simply converts high data to a low frequency
; and low data to a high frequency. The analog voltage is generated by
; a 1-pin PWM D/A. This program could be improved by adding a UART to
; the program, and sending the data out after it has been received. Using
; this method, the data can be manipulated.
;
; History: 1.01 & 1.02: Documentation Updates.
;
; INPUTS:
; RS-232 input on rx_pin (ra.1)
;
; OUTPUTS:
; PWM FSK signal on pwm_pin (ra.0)
; Echoed rs_232 characters on tx_pin (ra.2)
; LED flashes while transmitting on led_pin (rb.0)
;
; RESOURCES:
; Program memory: 72 Words
; Data memory: 8 bytes
; I/O Count: 3-5
;*****
; device pins28,pages1,banks8 ; 28-pin device, 1 pages, 8 banks of RAM
; device oschs,turbo,optionx,stackx ; High speed oscillator, turbo mode,
; ; option register extend, 8-level stack

```

```

        freq    50_000_000                ; default run speed = 50MHz
        ID      'FSK_TX10'                ; Version = 1.01

        reset   start                      ; JUMP to start label on reset
;*****
; Watches (For Debug in SX_Key software V.1.0 +)
;*****
        watch   freq_acc_low,16,uhex
        watch   freq_count_low,16,uhex
        watch   sin,8,sdec
        watch   sinvel,8,sdec
        watch   pwm0,8,udec
        watch   pwm0_acc,8,udec

;*****
; Macros (These only required on SX DTMF DEMO board)
;*****
enable_o    macro    0
            clrb     in_out
            endm

;*****

disable_o    macro    0
            setb     in_out
            endm

;*****

;*****
; Equates for common data comm frequencies
;*****
SX_DTMF_DEMO

f697_h      equ     $012    ; DTMF Frequency
f697_l      equ     $09d

f770_h      equ     $014    ; DTMF Frequency
f770_l      equ     $090

f852_h      equ     $016    ; DTMF Frequency
f852_l      equ     $0c0

f941_h      equ     $019    ; DTMF Frequency
f941_l      equ     $021

```

```

f1209_h      equ    $020    ; DTMF Frequency
f1209_l      equ    $049

f1336_h      equ    $023    ; DTMF Frequency
f1336_l      equ    $0ad

f1477_h      equ    $027    ; DTMF Frequency
f1477_l      equ    $071

f1633_h      equ    $02b    ; DTMF Frequency
f1633_l      equ    $09c

f1300_h      equ    $022    ; 1300Hz Signifies HIGH data in Bell202 Spec
f1300_l      equ    $0b7

f2100_h      equ    $038    ; 2100Hz Signifies LOW data in Bell202 Spec
f2100_l      equ    $015

```

```

;*****
; Pin Definitions (These definitions are for SX DTMF DEMO boards)
;*****

```

```

PWM_pin      equ    ra.0    ; Pin used for PWM output
rx_pin       equ    ra.1    ; Pin used for rs-232 receive
tx_pin       equ    ra.2    ; Pin used for rs-232 transmit
in_out       equ    ra.3    ; enables the output on the SX Modem Demo's
hook         equ    rb.4    ; Picks up the line on SX Modem Demo's
led_pin      equ    rb.0    ; The LED on SX Modem Demo's

```

```

;*****
; Bank 0 Variables
;*****

```

```

                org    $10

sin_gen_bank   =      $

freq_acc_low   ds      1      ; 16-bit accumulator which decides when to increment the sine wave
freq_acc_high  ds      1      ;
freq_count_low ds      1      ; 16-bit counter which decides which frequency for the sine wave
freq_count_high ds     1      ; freq_count = Frequency * 6.83671552
sin            ds      1      ; The current value of the imitation sin wave
sinvel         ds      1      ; The velocity of the sin wave

PWM_bank       =      $

pwm0_acc       ds      1      ; PWM accumulator
pwm0           ds      1      ; current PWM output

```

```

;*****
; Interrupt
;
; With a retiw value of -163 and an oscillator frequency of 50MHz, this
; code runs every 3.26us.
;*****
        org      0
;*****
PWM_OUTPUT
; This outputs the current value of pwm0 to the PWM_pin. This generates
; an analog voltage at PWM_pin after filtering
;*****
        bank     PWM_bank
        add       pwm0_acc,pwm0          ; add the PWM output to the accumulator
        snc
        jmp       :carry                  ; if there was no carry, then clear
                                           ; the PWM_pin

        clrb      PWM_pin
        jmp       PWM_out

:carry
        setb      PWM_pin                  ; otherwise set the PWM_pin

PWM_out
;*****
sine_generator
; This routine generates a synthetic sine wave with values ranging
; from -32 to 32. Frequency is specified by the counter. To set the
; frequency, put this value into the 16-bit freq_count register:
; freq_count = FREQUENCY * 6.83671552 (@50MHz)
;*****

        bank     sin_gen_bank

        add       freq_acc_low,freq_count_low;2 ;advance sine at frequency
        jnc       :no_carry                ;2,4 ; if lower byte rolls over
        inc       freq_acc_high            ; carry over to upper byte
        jnz       :no_carry                ; if carry causes roll-over
        mov       freq_acc_high,freq_count_high ; then add freq counter to accumulator (which should be zero,
                                           ; so move will work)
                                           ; and update sine wave

        jmp       :change_sin

:no_carry
        add       freq_acc_high,freq_count_high ; add the upper bytes of the accumulators
        jnc       :no_change

:change_sin

        mov       w,++sinvel               ;1 ; if the velocity of sine
        sb        sin.7                    ;1 ; is positive, accelerate

```

output - but - low
.. high

```

        mov     w,--sinvel      ;1          ; it. Otherwise, decelerate it.
        mov     sinvel,w        ;1
        add     sin,w           ;1          ; add the velocity to sin

        mov     pwm0,sin        ; mov the value of SIN into the PWM output
        add     pwm0,#128       ; add 128 to put it in the center of the PWM output

:no_change

sine_generator_out
;*****
;ISR_DONE
; This is the end of the interrupt service routine. Now load 163 into w and
; perform a retiw to interrupt 163 cycles from the start of this one.
; (3.26us@50MHz)
;*****

        mov     w,#-163         ;1          ; interrupt 163 cycles after this interrupt
        retiw   ;3              ; return from the interrupt
;*****
;
;*****
; Main program
;*****
start                                     ; Program goes here on power-up.

                                     ; First, Initialize the software

        bank    sin_gen_bank
        mov     sin,#32         ; init variables. A sine starts at 1, a cos wave starts at 0.
        mov     sinvel,#0
;
        mov     sin,#-4         ; use these values for a wave which is 90 degrees out of phase.
;
        mov     sinvel,#-8

        mov     ra,#%1111       ; make tx_pin high, disable the output, and put a 1 on the PWM output
        mov     rb,#%11101110   ; no hook, and no LED on
        mov     !ra,#%0010      ; ra0 = output (pwm), ra1 = input (rx) ra2 = output (tx) ra3 = in_out (output

enable)

        mov     !rb,#%11101110

        mov     m,#$D           ; make ra0 cmos-level
        mov     !ra,#%11110
        mov     m,$F

        jb      rx_pin,$        ; wait until data is received (Start bit is a low)

```



```

mov    !option, #%00011111      ; enable wreg and rtcc interrupt

setb   hook                      ; pick up the line (required on DTMF DEMO boards)
enable_o                      ; enable the output (required on DTMF DEMO boards)

```

```
main_loop
```

```

;*****
; Main_Loop loops forever and changes the output frequency
; of the sine wave depending on the state of the rx_pin
;*****

```

```

:rx_is_low    jb      rx_pin, :rx_is_high      ; If the rs_232 receive pin is currently a low
               setb   led_pin                  ; Turn the LED on
               clrb   tx_pin                    ; Echo the character
               mov     freq_count_high, #f2100_h ; output a frequency of 2100Hz
               mov     freq_count_low, #f2100_l
               jmp     main_loop
:rx_is_high   clrb   led_pin                  ; Else turn the LED off
               setb   tx_pin                    ; turn Echo the character
               mov     freq_count_high, #f1300_h ; output a frequency of 1300Hz
               mov     freq_count_low, #f1300_l
               jmp     main_loop

```

```

;*****

```

```

; Copyright © 1998 Scenix Semiconductor, Inc. All rights
; reserved.

```

```

; Scenix Semiconductor, Inc. assumes no responsibility or liability for
; the use of this [product, application, software, any of these products].

```

```

; Scenix Semiconductor conveys no license, implicitly or otherwise, under
; any intellectual property rights.
; Information contained in this publication regarding (e.g.: application,
; implementation) and the like is intended through suggestion only and may
; be superseded by updates. Scenix Semiconductor makes no representation
; or warranties with respect to the accuracy or use of these information,
; or infringement of patents arising from such use or otherwise.

```

```

; Scenix Semiconductor products are not authorized for use in life support
; systems or under conditions where failure of the product would endanger
; the life or safety of the user, except when prior written approval is
; obtained from Scenix Semiconductor.

```

```

;*****

```

10281
 10281 1113
 10281